An analogy based approach for solving target sense verification

Georgios Zervakis, Emmanuel Vincent, Miguel Couceiro, Marc Schoenauer, Esteban Marquer

An Analogy based Approach for Solving Target Sense Verification

Georgios Zervakis
georgios.zervakis@inria.fr
Université de Lorraine, CNRS, INRIA, LORIA
Nancy, France

Emmanuel Vincent
emmanuel.vincent@inria.fr
Université de Lorraine, CNRS, INRIA, LORIA
Nancy, France

Miguel Couceiro
miguel.couceiro@inria.fr
Université de Lorraine, CNRS, INRIA, LORIA
Nancy, France

Marc Schoenauer
marc.schoenauer@inria.fr
INRIA TAU, LISN (CNRS & Univ. Paris-Saclay)
France

Esteban Marquer
esteban.marquer@inria.fr
Université de Lorraine, CNRS, INRIA, LORIA
Nancy, France

ABSTRACT

Contextualized language models have emerged as a de facto standard in natural language processing due to the vast amount of knowledge they acquire during pretraining. Nonetheless, their ability to solve tasks that require reasoning over this knowledge is limited. Certain tasks can be improved by analogical reasoning over concepts, e.g., understanding the underlying relations in *"Man is to Woman as King is to Queen"*. In this work, we propose a way to formulate target sense verification as an analogy detection task, by transforming the input data into quadruples. We present AB4TSV (*Analogy and BERT for TSV*), a model that uses BERT to represent the objects in these quadruples combined with a convolutional neural network to decide whether they constitute valid analogies. We test our system on the WiC-TSV evaluation benchmark, and show that it can outperform existing approaches. Our empirical study shows the importance of the input encoding for BERT. This dependence gets alleviated by integrating the axiomatic properties of analogies during training, while preserving performance and improving interpretability.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**.

KEYWORDS

analogical reasoning, BERT, target sense verification

ACM Reference Format:
Georgios Zervakis, Emmanuel Vincent, Miguel Couceiro, Marc Schoenauer, and Esteban Marquer. 2022. An Analogy based Approach for Solving Target Sense Verification. In *Proceedings of 6th International Conference on Natural Language Processing and Information Retrieval (NLPIR) (NLPIR 2022).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

NLPIR 2022, December 16–18, 2022, Bangkok, Thailand

ACM ISBN 978-1-4503-9763-6…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recent efforts have focused in understanding how semantic and syntactic knowledge could be encoded in pretrained language models (PLMs) [1–5]. Despite their great success in a plethora of downstream tasks, the ability of these models to perform reasoning is limited and understudied [6, 7]. Designing PLMs that are able to reason over the knowledge they possess may not necessarily translate into better performance, however it is a prerequisite for interpretability [8]. The type of reasoning varies depending on the task at hand, e.g., answering chronological questions and summarizing events (temporal reasoning), selecting the most plausible explanation given a set of observations and hypotheses (abductive reasoning), understanding whether the meaning of a given text entails that of another (semantic inference), or finding common relations between pairs of words or phrases (analogical reasoning).

Analogical reasoning is one of the most used inference approaches in everyday life since it can be easily adapted to many commonsense applications involving reasoning: problem solving, modeling, planification, etc. Analogies also constitute a natural approach to modeling medical reasoning as practiced by physicians and medical staff. Further applications are found in natural language processing in tasks such as machine translation [9], visual question-answering [10], semantic [11] and morphological [12] problems.

Solving analogy-based problems requires the system to learn how to reason over relations of the form $A : B :: C : D$, which we read as "*A is to B as C is to D*". Following the axiomatization from [13], a quaternary relation constitutes an analogical proportion if the following properties hold true: $\forall A,\ B,\ C,\ D$,

(1) $A : B :: C : D \Rightarrow C : D :: A : B$ (symmetry)
(2) $A : B :: C : D \Rightarrow A : C :: B : D$ (central permutation)

Analogies combined with data-driven methods were proven to be beneficial in a variety of tasks ranging from transfer learning to data augmentation and explainable AI [14–18]. However, to the best of our knowledge, they have not been applied to word sense disambiguation (WSD) tasks.

Target Sense Verification (TSV) [19] is a WSD task in which the system is provided with a target word in context on the one hand and a definition and a set of hypernyms of that word on the other hand, and it must decide whether their senses match or not. Consider for instance the context *"home is where the heart is"*, the definition *"where you live at a particular time"* and the set

of hypernyms *"residence, abode"*, all corresponding to the target word *home*, as shown in Figure 1. To disambiguate the meaning of *home* in that specific context, one must compare and reason about the underlying relations between these concepts, e.g., infer that *home* in this context refers to an environment rather than a place as conveyed by the definition and the hypernyms.

Although TSV is not originally viewed as an analogy problem, analogical reasoning can be applied to solve it. To illustrate this we focus on two contexts – in one the target word *home* matches the target sense given by the definition and the hypernyms, whereas in the other it does not (Figure 1). Observe that the definition and the hypernyms always correspond to the same sense S_T (target sense). We denote this relation by R'. Solving TSV requires us to find whether the sense of *home* in context S_I (intended sense) corresponds to S_T. Let R be the relation that *home* points to S_I and hypernyms[1] point to S_T. We can now reformulate the problem in the form of analogical proportions such as, e.g., *home : hypernyms :: definition : hypernyms*, and check whether it constitutes a valid analogy in each context or not (analogy detection). If R is approximately the same as R', then their senses match and we output True, otherwise we output False. This modification essentially allows us to reuse PLMs that are suitable for solving TSV, and at the same time it gives us access to existing tools for tackling analogies. This way we can test if the combination of both is beneficial in terms of performance on the task and in terms of interpretability of the hybrid model. Furthermore, we focus on TSV because it is more complex than classical word analogy or lexical relation classification tasks [16]. Indeed, TSV does not compare isolated words but words in context, lists of words and full sentences.

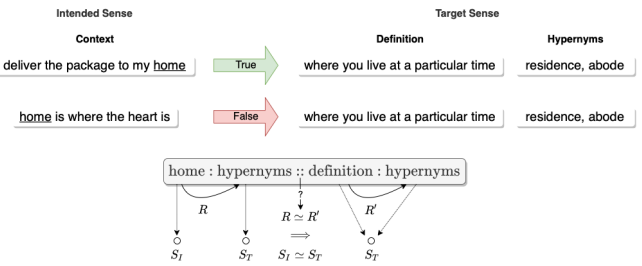

Figure 1: Illustration of translating TSV into analogy detection.

In this paper, we tackle TSV in terms of analogical reasoning by combining BERT [20] with a convolutional neural network (CNN) architecture that models analogical proportions by design and that is used for detecting analogies [11, 12]. Our experiments demonstrate that the position of the definition and hypernyms in the BERT input sequence, as well as the emphasis on the hypernyms, significantly impact the final performance. Moreover, we achieve competitive results on the WiC-TSV evaluation benchmark [19]. Finally, enforcing analogical reasoning by using the axiomatic properties of analogical proportions explicitly during training, yields comparable performance and alleviates the dependence on the input encoding of BERT, rendering our model more interpretable.

[1]The definition can also be used but for the sake of this example we use the hypernyms.

The main contributions of this work are the following:

- we reformulate TSV problem as an analogy detection problem,
- we propose AB4TSV, a hybrid approach for solving TSV,
- we optimize the input encodings for AB4TSV,
- we demonstrate that AB4TSV achieves competitive performance on the WiC-TSV evaluation benchmark,
- we show empirically that enforcing the axiomatic properties of analogies during training yields a more interpretable model.

The paper is organized as follows. We discuss related work in Section 2. The model architecture is described in Section 3, and the experimental setup in Section 4. In Section 5 we analyze the results, and we discuss conclusions and perspectives for future work in Section 6.

2 RELATED WORK

2.1 Approaches for solving TSV

Breit et al. [19] propose an architecture for TSV that consists of BERT followed by a fully-connected, binary classifier layer. The classifier uses the embeddings of the `[CLS]` token, the target word, and the definition and/or the hypernyms as inputs. The authors tested both its base and large versions referred to as BERT-B and BERT-L, respectively. They compare these with FastText embeddings and two unsupervised baselines based on BERT (U-BERT) and DistilBERT (U-dBERT). Similarly, Moreno et al. [21] fine-tune two distinct BERT models on the hypernyms and the definition, using the `[CLS]` token embedding for classification, and they aggregate the two classifier outputs at inference time. Vandenbussche et al. [22] ran an extensive study of BERT for TSV, including data augmentation, freezing the model parameters during fine-tuning, applying different pooling strategies to obtain the classifier input, and masking the target word in the context. Liu et al. proposed a more generic approach called MIRROWIC [23] and tested it on various lexical semantic tasks including TSV. This fully unsupervised approach based on *contrastive learning* aims to extract improved word embeddings from PLMs such as BERT. Specifically, they create positive and negative pairs for a target word by making use of augmentation, masking and dropout techniques, as well as raw samples from Wikipedia. Then, they train the embeddings such that positive pairs are pulled closer, while negative pairs are pushed apart. To evaluate their method on TSV, they constructed manual templates involving the target word, the definition and/or the hypernyms, and compared the cosine similarities of the target word embeddings in the original context and the template.

2.2 Embedding systems for detecting analogies

Analogical reasoning with distributional word embeddings was first discussed by Mikolov et al. [24]. They showed that such vectors can model relations in the data through vector differences such that if A, B, C, D are in analogical proportion, then the differences of their respective embeddings[2] $(\mathbf{B} - \mathbf{A})$ and $(\mathbf{D} - \mathbf{C})$ ought to be similar. More recently, Ushio et al. distilled relation embeddings between word pairs directly from BERT [16]. To that end, they fine-tuned

[2]We denote by boldface $\mathbf{A}$ the embedding of object A.

BERT such that the embeddings of word pairs belonging to the same relation class are closer than those belonging to a different class. Their method outperformed state-of-the-art methods on several analogy and relation classification benchmarks. In a following study [25], they assessed the extent to which PLMs are capable of solving analogies without further fine-tuning. Their results demonstrate that PLMs can detect analogies but are sensitive to the choice of the hyperparameters. Furthermore, they show that such models are limited when it comes to more complex relations, and often perform worse than traditional word embedding systems. Afantenos et al. [15] attempted to identify analogical proportions between sentences. To this end, they proposed a more relaxed definition of analogical proportions that is better suited for sentences by substituting the central permutation property with that of *internal reversal*. They obtained promising results. Lim et al. [11] proposed a CNN architecture for detecting semantic analogies, where they frame the task as an image classification problem. CNNs are good at capturing high level features in pictures, hence they detect analogical proportions by stacking the embeddings of **A**, **B**, **C** and **D** into an image and feeding it in the CNN. In the same spirit, Alsaidi et al. [12] adapted the previous model to detect morphological analogies in different languages. Training a character-based CNN with data augmentation using the properties of analogical proportions, they outperformed results of state-of-the-art symbolic approaches ([26, 27]).

3 AB4TSV ARCHITECTURE

Following [11, 12] we make use of a CNN classifier that explicitly models the relations *"is to"* and *"as"* in the analogy, as an interpretable alternative to the black-box classifier of [19]. The architecture of the model is composed of two main parts. First, we encode the words or phrases to be compared (target word, context, definition, hypernyms, etc.) into 4 embeddings **A**, **B**, **C**, **D** using the final encoder layer of BERT. Then we stack them into an $n \times 4$ matrix, where n is the embedding size. This matrix serves as input to a CNN classifier that consists of the following layers:

- A convolutional layer with 128 filters of size 1×2 with stride $(1, 2)$ followed by a ReLU activation. The output of this operation is an $128 \times n \times 2$ matrix. Intuitively, this layer models "*is to*" in *"A is to B"* and *"C is to D"*.
- A convolutional layer with 64 filters of size 2×2 with stride $(2, 2)$ followed by a ReLU activation. The output of this operation is flattened into a vector of length $64 \times (n-1)$. Intuitively, this layer models the relation "*as*" in *"A is to B as C is to D"*.
- A fully-connected layer followed by a sigmoid activation with a scalar output.

The main difference with [11, 12] is the shifting from static to contextualized embeddings. While they use pre-trained GloVe vectors or train a character-based CNN to extract word representations, we utilize BERT to extract **A**, **B**, **C**, **D**. This, together with our input encoding optimization (Section 3.2), allows us to efficiently compare objects of different structure such as words, list of words, full sentences and/or their combinations, rather than just isolated words. The proposed AB4TSV architecture is depicted in Figure 2.

3.1 Choice of Analogical Relation

Tackling TSV based on analogical reasoning requires us to select the appropriate $A, B, C, D \in S$ such that the relation $A : B :: C : D$ yields good classification performance. Let $S = \{cls, tgt, ctx, def, hyps, descr\}$ be the set of tokens that can be obtained from BERT[3]. The selection of $tgt, ctx, def, hyps$ as possible candidates for A, B, C, D is essential, since these are the main components that carry the senses to be compared according to the task. Additionally, the embedding of *cls* can generally be seen as a representation of the whole input, therefore it may capture key information both from context, and definition/hypernyms. Finally, inspired by the authors of WiC-TSV [19] we also test for *descr*, which essentially treats definition and hypernyms as a whole rather than separate units. This selection of S allows us to compare relevant instances of different structure (special tokens, words in context, list of words, full sentences or their combinations), and thus, makes the task more challenging than typical word analogy or lexical relation classification.

3.2 Input Encoding Selection

The order in which the context, definition, and hypernyms are fed into BERT, has a direct impact on the embeddings of the A, B, C, D candidates in S. Preliminary experiments using the (context, definition-hypernyms) input encoding format illustrated in Figure 2, have shown that most relations seem to work except when *hyps* is included. This may be due to the way hypernyms are encoded: they are simply a set of words separated by commas, appended to the definition. BERT was originally trained on syntactically correct sentences, hence it might find them difficult to interpret. Therefore, we introduce alternative ways of encoding the definition and the hypernyms into embeddings based on the following operations:

- **swap**: exchanging the position of the definition and the hypernyms in the input sentence of BERT;
- **fc**: enclosing the hypernyms with focus characters, e.g., *residence, abode* becomes *$ residence, abode $*;
- **em**: enclosing the hypernyms with entity markers, e.g., *residence, abode* becomes `[H]` *residence, abode* `[/H]`;

Following [19] we always apply **fc** to the target word in the context. The use of focus characters/entity markers works as a form of weak supervision, pointing out important terms in the sentence [28, 29]. This does not directly address the syntactic correctness issue caused by the hypernyms, however, it instructs BERT to treat them in a special way. That is, all hypernyms will now share a common characteristic in the data that will, ideally, alter their embeddings in such a way that they become more meaningful for solving the task. Moreover, as shown by [29] part of the properties of employing such strategies, can be captured in the embeddings of the focus characters/entity markers themselves. Therefore, we include them in the computation of **hyps**, as a means to explicitly transfer these properties directly to the representation of hypernyms.

Based on these three operations, we tested the following 6 input encodings:

[3] **cls**: embedding of the `[CLS]` token,
tgt: embedding of the target word,
ctx: average of the embeddings of all words in the context,
def: average of the embeddings of all words in the definition,
hyps: average of the embeddings of all hypernyms,
descr : average of the embeddings of all words in the definition and all hypernyms.

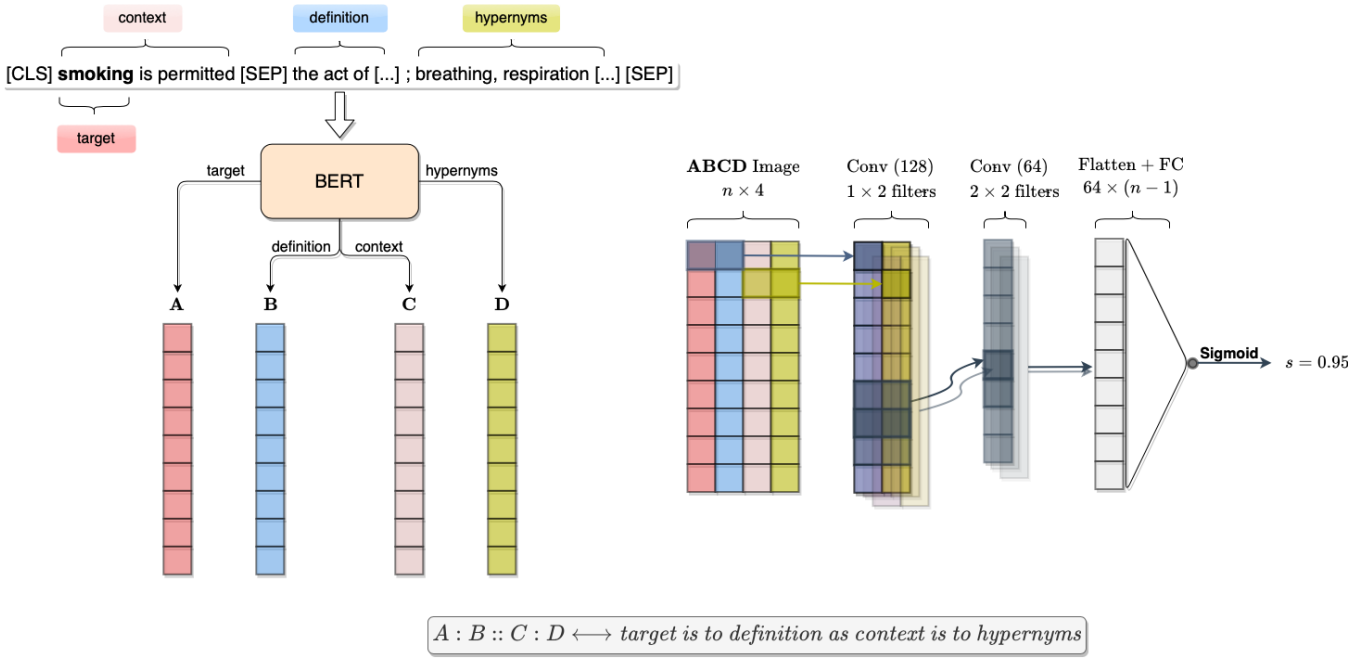

Figure 2: Overview of AB4TSV. In this example we want to test whether $A : B :: C : D$ is a valid analogy when A = target, B = definition, C = context *and* D = hypernyms. The inputs are patched into a pair of sentences (context, definition-hypernyms) and fed into BERT. The embeddings of A, B, C, D extracted from the last encoder layer of BERT are stacked into an $n \times 4$ matrix and input to the CNN classifier. Here, the classifier outputs *True* since the output value $s = 0.95 > \delta$, where the decision threshold δ is set to 0.5.

default: `[CLS] context [SEP] definition; hypernyms [SEP]`

default+fc: `[CLS] context [SEP] definition; $ hypernyms $ [SEP]`

default+em: `[CLS] context [SEP] definition; [H] hypernyms [/H] [SEP]`

swap: `[CLS] context [SEP] hypernyms; definition [SEP]`

swap+fc: `[CLS] context [SEP] $ hypernyms $; definition [SEP]`

swap+em: `[CLS] context [SEP] [H] hypernyms [/H]; definition [SEP]`

4 EXPERIMENTAL SETUP

In this section, we first present the data used to train and evaluate our system. Then, we describe the experimental procedure along with some technical details concerning implementation, evaluation and baselines for comparison.

4.1 Data

We use the Words-in-Context-TSV (WiC-TSV) dataset, which was designed specifically for TSV [19]. The data are pairs of context and definition-hypernyms, where the definition and hypernyms correspond to the same sense of the target word. General-domain instances are extracted from WordNet and Wiktionary (WNT/WKT). Domain-specific instances for Cocktails (CLT) and Medical Subjects (MSH) were taken from "All about cock-tails"[4] and MeSH[5] thesauri respectively, while Computer Science (CPS) examples were manually constructed. The training and development sets include general-domain sentences only, while the test set includes domain-specific sentences too (see Tables 1 & 2). The task is divided into three sub-problems taking into account only the definition (sub-task 1), only the hypernyms (sub-task 2), or both (sub-task 3). In the following we focus on sub-task 3 only.

Table 1: Statistics of the WiC-TSV dataset. The $\mathcal{P}_+$ column refers to the percentage of positive examples.

		Total	$\mathcal{P}_+$
Train	**WNT/WKT**	2137	0.56
Dev	**WNT/WKT**	389	0.51
Test	**WNT/WKT**	717	0.54
	Domain-specific	589	0.47
	MSH	205	0.52
	CTL	216	0.43
	CPS	168	0.46
	All	1306	0.51

[4]http://vocabulary.semantic-web.at/cocktails
[5]https://www.nlm.nih.gov/mesh/meshhome.html

Table 2: WiC-TSV samples taken from the development set. The target word in the context is highlighted in bold. Notice that in some examples the meaning of the target word can be easily disambiguated solely from the context, whereas in others the definition and/or hypernyms are necessary.

Context	Definition	Hypernyms	Label
A **marriage** of ideas.	A close and intimate union.	union, unification	True
A **fight** broke out at the hockey game.	The act of fighting; any contest or struggle.	conflict, struggle, battle	True
My neighbor was the lead **role** in last year's village play.	The actions and activities assigned to or required or expected of a person or group.	duty	False
They went bankrupt during the economic **crisis**.	A crucial stage or turning point in the course of something.	juncture, occasion	False

4.2 Analogical Proportions Optimization

Regarding the number of combinations of $A, B, C, D \in S$, there are $|S|^4 = 6^4 = 1,296$ possible relations in total. To ensure that we test whether the intended sense of the target word in the context corresponds to the target sense in the definition/hypernyms, we first distinguish between two sets: $S_1 = \{cls, tgt, ctx\}$ and $S_2 = \{cls, def, hyps, descr\}$ where $S_1 \cup S_2 = S$. The former set includes embeddings which primarily contain information coming from the context, while the latter involves embeddings which reflect the information found in the definition-hypernyms. *cls* belongs to both since it represents the whole input. Next, we define the following rules: $\forall A, B, C, D \in S$

$$A \neq B \vee C \neq D$$

$$\neg\Big[\big[(A \in S_1 \setminus S_2) \wedge (B, C, D \in S_1)\big] \vee \big[(A \in S_2 \setminus S_1) \wedge (B, C, D \in S_2)\big]\Big]$$

The first rule ensures that we avoid relations where embeddings on either side are identical, e.g., *A is to A as C is to D*. The second rule makes sure that each relation contains embeddings from both sources of information. In other words, A, B, C, D cannot be instantiated from S_1 or S_2 exclusively. These rules reduce the number of relations to be tested to 768. For each choice of input encoding and relation, we train our system 4 times using different random seeds, resulting in $6 \times 768 \times 4 = 18,432$ runs. A single run takes approximately 35 minutes on 1 Nvidia GTX 1080 Ti 11GB.

4.3 Assessing and Enforcing Invariance to the Permutations of Analogical Proportions

A key part of our experiments is to employ the permutation properties of analogical proportions (see Section 1) to assess (i) whether analogical reasoning is beneficial in terms of performance on the task, and (ii) whether the model naturally learns to be invariant to these permutations or they must be explicitly enforced at training time. To assess whether the model is invariant, we compute the embeddings **A**, **B**, **C**, **D** of the given relation to be tested, and simply compare the performance achieved when feeding the initial relation vs. the relations obtained by permuting analogical proportions to the classifier. To enforce permutation invariance at training time, we include both the initial and the permuted relations in each mini-batch. Note that reflexivity ($A : B :: A : B$) is discarded since it does not take C, D into account. We distinguish training without/with permutation invariance by adding a subscript pi, i.e., AB4TSV and AB4TSV$_{pi}$, respectively.

4.4 Technical Details

As baselines we consider the BERT variants published by [19], namely HyperBertCLS and HyperBert3. Both models are composed of BERT with a linear layer on top as a classifier. The key difference between them is that the former takes the embedding of the `[CLS]` token as input to the classifier, while the latter takes not only the embedding of `[CLS]` but also the embedding of the target word and the average of all words in the definition and all hypernyms.

Performance is measured in terms of accuracy and $F1$-score. Unless otherwise stated, all tables report statistics (mean and standard deviations) of these two measures over several independent runs. Since the test set labels of WiC-TSV are not publicly available, all reported scores on the test set are computed by the organizers [19].

The WiC-TSV data, scripts for tranining/evaluating AB4TSV and the baseline models, and source code for reproducing our experimental results are available at our GitHub repository[6].

5 RESULTS

In this section we present the results obtained from the optimization of the input encoding and the analogical relation, compare those with the baselines on the development set, and with existing approaches on the test set. Next, we discuss the effects of utilizing (or not) the analogical properties explicitly during the training process.

5.1 Impact of the Input Encoding

Figure 3 shows the mean accuracy achieved across all 4 runs sorted in ascending order for each input encoding and each analogical relation (Section 3.2). Overall, for all input encodings there exist some A, B, C, D combinations that result in good performances. However, some of them appear to be more sensitive than others to the selection of the analogical relation. In particular, swapping or not the position of the definition and the hypernyms in the sentence (plain lines), results in a large amount of bad accuracies for several A, B, C, D combinations ($0 \sim 400$ on the x-axis). Taking a closer look at these combinations we observe that all of them have at least one of A, B, C, D instantiated as `hyps`. This comes in agreement with our preliminary experiments, demonstrating that BERT is struggling to make sense out of the hypernyms probably because of their structure – a list of words that comes before or after the definition,

[6] https://github.com/gonconist/ab4tsv

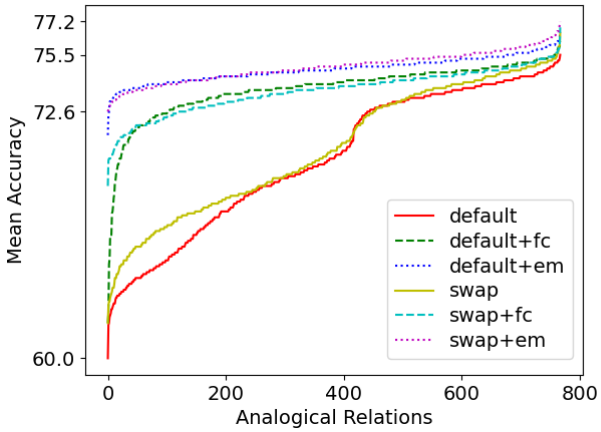

Figure 3: Mean accuracy achieved on the development set. Each curve represents one possible input encoding, and the horizontal axis represents the 768 possible relations $A : B :: C : D$ sorted in order of increasing accuracy.

and does not form a syntactically correct sentence. However, this effect is not present when we apply the **fc/em** strategies (dashed and dotted lines). Based on some additional experiments, we observed that simply enclosing the hypernyms with focus characters or entity markers at the input level, is not sufficient to alleviate this problem. What really makes a difference, is the inclusion of the embeddings of these special characters when computing the vector representation of the hypernyms. Finally, the entity markers strategy appears to consistently outperform the focus characters one on the development set. One possible explanation is that the same focus characters are also enclosing the target word in the context. Hence, the part of the input that comes after the target word and before the hypernyms is also enclosed in these special characters, possibly bringing some confusion to the model.

5.2 Comparison with other methods for TSV

We retrain the AB4TSV models with the best relation for each selected encoding, and report the results over 10 runs, as well as those of the baselines, in Table 3. The results clearly show that our system outperforms the baseline models. This holds true even for one-to-one comparisons on the respective encodings, further demonstrating the importance of the input encoding for BERT. Specifically, exchanging the position of the definition and the hypernyms (**swap**) and using focus characters/entity markers (**fc/em**) yields the best performance in terms of accuracy. Note also that `cls` is present in all analogical relations: This suggests that the `[CLS]` token is particularly important for solving this task.

Based on the results of the previous experiment we select the relation that performed best for each encoding, and train AB4TSV model 10 times by enforcing invariance to the permutations of analogical proportions at training time. The results, reported in Table 4, show a slight decrease in performance compared to the systems trained without the permutations of analogical proportions

Table 3: Accuracy and $F1$-score achieved on the development set by the proposed method and the two baselines. The Analogy column shows the relation that yielded the best performance for a given encoding.

Encoding	Analogy	Dev *Acc*	Dev *F1*
default	$cls : descr :: cls : ctx$	74.5 ± 0.015	77.0 ± 0.016
default+fc	$cls : def :: ctx : cls$	74.9 ± 0.010	77.3 ± 0.006
default+em	$tgt : descr :: cls : def$	75.4 ± 0.027	**77.8** ± 0.023
swap	$def : cls :: cls : ctx$	75.4 ± 0.016	77.7 ± 0.016
swap+fc	$def : ctx :: cls : hyps$	**75.8** ± 0.013	77.7 ± 0.013
swap+em	$hyps : def :: cls : ctx$	**75.8** ± 0.017	77.7 ± 0.012
	Baselines		
default		74.4 ± 0.014	**77.2** ± 0.009
default+fc		73.5 ± 0.027	75.2 ± 0.035
default+em	HyperBertCLS	74.0 ± 0.022	76.1 ± 0.019
swap		72.6 ± 0.028	74.4 ± 0.031
swap+fc		73.1 ± 0.028	75.2 ± 0.031
swap+em		**74.6** ± 0.024	76.6 ± 0.022
default		74.0 ± 0.014	76.9 ± 0.007
default+fc		73.9 ± 0.018	76.3 ± 0.018
default+em	HyperBert3	73.1 ± 0.031	75.2 ± 0.032
swap		73.8 ± 0.015	76.3 ± 0.015
swap+fc		73.5 ± 0.011	75.6 ± 0.013
swap+em		74.4 ± 0.011	75.7 ± 0.024

in Table 3, in the order of 1% absolute. However, notice that after enforcing analogical reasoning, the results are consistent across all encodings. This suggests that the model learns to be invariant to the input encoding, and thus, more interpretable.

Table 4: Accuracy and $F1$-score achieved on the development set by enforcing invariance to the permutations of analogical proportions at training time. The Analogy column shows the relation that yielded the best performance for a given encoding in the previous experiment.

Encoding	Analogy	Dev *Acc*	Dev *F1*
default	$cls : descr :: cls : ctx$	74.3 ± 0.016	76.1 ± 0.014
default+fc	$cls : def :: ctx : cls$	74.6 ± 0.008	76.6 ± 0.008
default+em	$tgt : descr :: cls : def$	**75.1** ± 0.014	**77.3** ± 0.013
swap	$def : cls :: cls : ctx$	74.2 ± 0.010	76.1 ± 0.011
swap+fc	$def : ctx :: cls : hyps$	74.8 ± 0.012	75.9 ± 0.024
swap+em	$hyps : def :: cls : ctx$	75.0 ± 0.009	76.4 ± 0.011

To assess the generalization capabilities of AB4TSV, we evaluate the performance of the best systems of the previous experiments on the test set. The results in Table 5 show that AB4TSV can outperform previously reported results on WiC-TSV both in accuracy and $F1$-score, according to whether the axiomatic properties of analogies are enforced at training time or not. This illustrates the usefulness of analogical reasoning to solve the task, even on specific instances that lie outside the training domain. Interestingly,

Table 5: Test set results of our best performing system trained with and without the permutations of analogical proportions, compared to previously reported results. All results are calculated by the authors of WiC-TSV benchmark [19].

Approach	Test *Acc*	Test *F*1
Supervised		
CTLR [21]	78.3	78.5
Vandenbussche et al. [22]	71.9	76.2
BERT-B [19]	76.6	78.2
BERT-L [19]	76.3	77.8
FastText [19]	53.4	63.4
AB4TSV+swap+em	75.7	77.5
AB4TSV+swap+fc	**78.6**	**79.8**
AB4TSV$_{pi}$+default+em	**78.6**	79.4
Unsupervised		
U-dBERT [19]	61.2	51.3
U-BERT [19]	60.5	51.9
MIRRORWIC [23]	73.7	–

swap+em that was the most dominant strategy in Figure 3, results in poorer performance compared to other approaches. After inspecting the detailed scores for this model, we observe that this decrease is present both in general and domain-specific examples on the test set: This particular selection of entity markers for the hypernyms does not generalize well on the test set. However, notice that AB4TSV$_{pi}$ using the same entity markers outperforms existing approaches, demonstrating once more that when trained using the axiomatic properties of analogies, AB4TSV becomes invariant to the input encoding selection, and thus more interpretable.

5.3 Invariance to the Permutations of Analogical Proportions

In order to measure the invariance of the model w.r.t. permutations, we focus on the relation that yielded the best accuracy on the analogical proportions optimization experiment (5$^{\text{th}}$ row of Table 3). In this case, we compare the performance obtained using the initial and the permuted analogical relations as input to the classifier, depending on whether we explicitly enforce invariance or not. Table 6 reports the results over 4 runs. As expected, the model is invariant to both permutations when we explicitly enforce it during training. Conversely, when we train on a single non-permuted relation, the performance for symmetry degrades a lot at test time, while that of central permutation decreases by a smaller margin. This makes sense for the specific relation ($def : ctx :: cls : hyps$) since central permutation ($def : cls :: ctx : hyps$) resembles more the original than symmetry ($ctx : hyps :: def : cls$).

6 CONCLUSION AND FUTURE WORK

In this work we proposed an alternative formulation for TSV based on analogical reasoning. More precisely, we compared directly the underlying relations between several components of the input text (target, context, definition, hypernyms), by developing a transformer-based architecture combined with a CNN classifier

Table 6: Results on the development set for the swap+fc encoding and the relation $def : ctx :: cls : hyps$. Permute column refers to enforcing (✓) or not (✗) invariance to the permutations of analogical proportions at training time.

Property	Permute	Dev *Acc*	Dev *F*1
base	✗	**76.2** ± 1.927	**78.0** ± 1.932
	✓	75.1 ± 1.611	76.8 ± 1.891
sym	✗	53.2 ± 17.10	61.4 ± 18.53
	✓	74.5 ± 1.949	76.4 ± 2.210
cp	✗	72.9 ± 3.596	73.4 ± 5.760
	✓	74.7 ± 2.104	76.5 ± 2.315

previously used for detecting analogies. The experimental results demonstrated the importance of the input encoding, suggesting that BERT is better off handling well-structured sentences or text that is specifically marked with special characters. Moreover, enforcing invariance w.r.t. the permutations of analogical proportions during training resulted in a more interpretable system that behaves consistently, irrespective of the input encoding, and performs comparably to its initial version (without the use of the permutations). Both approaches achieved competitive results on the WiC-TSV evaluation benchmark, displaying some generalization capabilities even for domain-specific examples outside of the training data. In the future, we plan to further investigate the usefulness of analogical reasoning on TSV by eliminating the contextualized dependence on the objects of interest, such as target word in context, definition, hypernyms, etc. To do so, we could split the current input into separate sentences and feed them into BERT independently. We also intend to borrow methodologies in natural language generation for solving analogies in the context of TSV, i.e., to generate X in the relation $A : B :: C : X$.

ACKNOWLEDGMENTS

This research was partially supported by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 952215 TAILOR and by the Inria Project Lab HyAIAI. Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

REFERENCES

[1] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, volume 1, pages 2227–2237, 2018.
[2] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019.
[3] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*, volume 1, pages 1073–1094, 2019.
[4] Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of BERT. In *NeurIPS*, volume 32, pages 8592–8600, 2019.
[5] Evan Hernandez and Jacob Andreas. The low-dimensional linear geometry of contextualized word representations. In *ACL*, pages 82–93, 2021.

[6] Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. oLMpics — on what language model pre-training captures. *TACL*, 8:743–758, 2020.

[7] A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in BERTology: What we know about how BERT works. *TACL*, 8:842–866, 2020.

[8] Ming Zhou, Nan Duan, Shujie Liu, and Heung-Yeung Shum. Progress in neural NLP: modeling, learning, and reasoning. *Engineering*, 6:275–290, 2020.

[9] Philippe Langlais, François Yvon, and Pierre Zweigenbaum. Improvements in analogical learning: Application to translating multi-terms of the medical domain. In *EACL*, pages 487–495, 2009.

[10] Julia Peyre, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Detecting unseen visual relations using analogies. In *ICCV*, pages 1981–1990, 2019.

[11] Suryani Lim, Henri Prade, and Gilles Richard. Solving word analogies: A machine learning perspective. In *ECSQARU*, volume 11726, pages 238–250, 2019.

[12] Safa Alsaidi, Amandine Decker, Puthineath Lay, Esteban Marquer, Pierre-Alexandre Murena, and Miguel Couceiro. A neural approach for detecting morphological analogies. In *DSAA*, pages 1–10, 2021.

[13] Yves Lepage. Analogy and formal languages. *Electron. Notes Theor. Comput. Sci.*, 53:180–191, 2004.

[14] Safa Alsaidi, Amandine Decker, Puthineath Lay, Esteban Marquer, Pierre-Alexandre Murena, and Miguel Couceiro. On the transferability of neural models of morphological analogies. In *AIMLAI*, volume 1524 of *CCIS*, pages 76–89, 2021.

[15] Stergos D. Afantenos, Tarek Kunze, Suryani Lim, Henri Prade, and Gilles Richard. Analogies between sentences: Theoretical aspects - preliminary experiments. In *ECSQARU*, volume 12897, pages 3–18, 2021.

[16] Asahi Ushio, José Camacho-Collados, and Steven Schockaert. Distilling relation embeddings from pre-trained language models. In *EMNLP*, pages 9044–9062, 2021.

[17] Mark T. Keane and Barry Smyth. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In *ICCBR*, volume 12311 of *LNCS*, pages 163–178. Springer, 2020.

[18] E. Hüllermeier. Towards analogy-based explanations in machine learning. In *MDAI*, volume 12256, pages 205–217, 2020.

[19] Anna Breit, Artem Revenko, Kiamehr Rezaee, Mohammad Taher Pilehvar, and José Camacho-Collados. WiC-TSV: An evaluation benchmark for target sense verification of words in context. In *EACL: Main Volume*, pages 1635–1645, 2021.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *HLT-NAACL*, pages 4171–4186, 2019.

[21] Jose G Moreno, Elvys Linhares Pontes, and Gaël Dias. CTLR@WiC-TSV: Target sense verification using marked inputs and pre-trained models. In *SemDeep-6*, pages 1–6, 2021.

[22] Pierre-Yves Vandenbussche, Tony Scerri, and Ron Daniel Jr. Word sense disambiguation with Transformer models. In *SemDeep-6*, pages 7–12, 2021.

[23] Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulic. MirrorWiC: On eliciting word-in-context representations from pretrained language models. In *CoNLL*, pages 562–574, 2021.

[24] Tomás Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *NAACL*, pages 746–751, 2013.

[25] Asahi Ushio, Luis Espinosa Anke, Steven Schockaert, and Jose Camacho-Collados. BERT is to NLP what AlexNet is to CV: Can pre-trained language models identify analogies? In *ACL-IJCNLP*, volume 1, pages 3609–3624, 2021.

[26] Pierre-Alexandre Murena, Marie Al-Ghossein, Jean-Louis Dessalles, and Antoine Cornuéjols. Solving analogies on words based on minimal complexity transformation. In *IJCAI*, pages 1848–1854, 2020.

[27] Rashel Fam and Yves Lepage. Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages. In *LREC*, pages 1060–1066, 2018.

[28] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *EMNLP-IJCNLP*, pages 3507–3512, 2019.

[29] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *ACL*, pages 2895–2905, 2019.